CS 312 Data Structures and Functional Programming
Fall 2005

Problem set 5

11/22/2005

This problem set is due December 1st, 2005.

1. We often say that an algorithm is "exponential in n" (where n is some measure of problem size), without saying what the base is. Is $3^n = O(2^n)$? Is $2^n = O(3^n)$?

2. Give asymptotic upper bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bound as tight as possible.

   - $T(n) = 2T(n/2) + n^3$
   - $T(n) = T(n-1) + log(n)$

3. Emil is studying the recurrence relation $f(n) = f(n-1) + f(n-2)$ for $n > 2$. Depending on the values you choose for $f(1)$ and $f(2)$, you get different series that satisfy the same recurrence relation. Emil asserts that one possible series that satisfies this recurrence relation is the geometric series r^n for $n = 1, 2, 3, 4..$, where r is a mystery value you have to find.

   (a) Find two possible values for r. The larger of the two values is called the *golden ratio*.

   (b) Argue that if two sequences r_1^n and r_2^n are both solutions to this equation, then any linear combination of these sequences is also a solution.

   (c) Let r_1 and r_2 be the two values that you found for r in the previous part, where $r_1 > r_2$. Find c_1 and c_2 such that the sequence $c_1 r_1^n + c_2 r_2^n$ is the Fibonacci sequence.

4. Prove the following generalization of Fibonacci problem. Let c_1 and c_2 be real numbers, and suppose that the equation $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2. Then the sequence $\{a_n\}$ is a solution to

the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ iff $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for $n = 1, 2, 3...$ where α_1 and α_2 are constants.

5. Prove the following. Let c_1 and c_2 be real numbers, and suppose that the equation $r^2 - c_1 r - c_2 = 0$ has only *one* root r_0. Then the sequence $\{a_n\}$ is a solution to the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ iff $a_n = \alpha_1 r_0^n + \alpha_2 n.r_0^n$ for $n = 1, 2, 3...$ where α_1 and α_2 are constants.

6. Use the result of the previous problem to solve the linear recurrence $a_n = 6a_{n-1} - 9a_{n-2}$ where $a_1 = 1$ and $a_2 = 6$.

7. For each of the following ML functions, draw an abstract syntax tree, decorate each edge with an appropriate type scheme, write down the set of type equations and solve them to find the most general type for the function. To help us understand your solution, you must follow the convention discussed in class in which the type identifier for an identifier v is t_v or tv or something similar.

   (a) `fun compose f g x = f(g(x))`

   (b)
   ```
   fun filter p l =
       if null(l)
       then nil
       else if p(hd(l))
            then hd(l)::filter p (tl(l))
            else filter p (tl(l))
   ```